The ten main components of a report in data science are

* Title of the Project
* Table of Contents
* Abstract/Project Summary
* Introduction
* Dataset description
* Methods and Algorithms
* Detailed Analysis
* Final results
* Conclusion
* References

Abstract:

In this project we are scraping the data for the data analytics, Using selenium, Mysql, and Streamlit, Using redbus website we are scrapping the data for the aims to revolutionize the transportation industry by providing a comprehensive solution for collecting, analyzing, and visualizing bus travel data. By utilizing Selenium for web scraping, this project automates the extraction of detailed information from Redbus, including bus routes, schedules, prices, and seat availability. By streamlining data collection and providing powerful tools for data-driven decision-making, this project can significantly improve operational efficiency and strategic planning in the transportation industry.

Introduction:

Identify the specific business problem or challenge that needs to be addressed through data analysis. Consider the importance of the problem and its potential impact on the overall business. Specify the desired outcome or objectives of the analysis.

In this we are going to get the data for analysis of the transportation industry, for that we are using the redbus website to extract the data using the data scraping using the selenium, Here we are getting the detailed data and we are storing the data in mysql database, To analyse the data and to visualize the data we are using the Streamlit

Data Set Description:

**Dataset is a set of data grouped into a collection** with which developers can work to meet their goals. In a dataset, the rows represent the number of data points and the columns represent the features of the Dataset. They are mostly used in fields like machine learning, business, and government to gain insights, make informed decisions, or train algorithms. Datasets may vary in size and complexity and they mostly require cleaning and preprocessing to ensure data quality and suitability for analysis or modeling.

The common types of data include:

* Text data
* Image data
* Audio data
* Video data
* Numeric data

Data Scrapping:

**is an automatic method to obtain large amounts of data from websites.** Most of this data is unstructured data in an HTML format which is then converted into structured data in a spreadsheet or a database so that it can be used in various applications. There are many different ways to perform web scraping to obtain data from websites. These include using online services, particular API’s or even creating your code for web scraping from scratch.

Scrapping Programatically:

We are using python and selenium for the data scraping

Ethics of Web Scraping

There are quite a few aspects of scraping that are ethically questionable [13].

A few examples of ethical issues regarding scraping are:

• Aggressively scraping can cause certain sites to slow down or even crash.

• Potentially reducing the value of a site as its information can now be

found elsewhere. This redirection of internet traffic might results in a

loss of monetization, through for example ads.

• The legal aspects of scraping are still a grey zone, thus can be taken

advantage of.

Data Scraping Techniques

Here are a few techniques commonly used to scrape data from websites. In general, all web scraping techniques retrieve content from websites, process it using a scraping engine, and generate one or more data files with the extracted content.

HTML Parsing

HTML parsing involves the use of JavaScript to target a linear or nested HTML page. It is a powerful and fast method for extracting text and links (e.g. a nested link or email address), scraping screens and pulling resources.

DOM Parsing

The Document Object Model (DOM) defines the structure, style and content of a typically use a DOM parser to view the structure of web pages in depth. DOM parsers can be used to access the nodes that contain information and scrape the web page with tools like XPath. For dynamically generated content, scrapers can embed web browsers like Firefox and Internet Explorer to extract whole web pages (or parts of them).

Vertical Aggregation

Companies that use extensive computing power can create vertical aggregation platforms to target particular verticals. These are data harvesting platforms that can be run on the cloud and are used to automatically generate and monitor bots for certain verticals with minimal human intervention. Bots are generated according to the information required to each vertical, and their efficiency is determined by the quality of data they extract.

XPath

XPath is short for XML Path Language, which is a query language for XML documents. XML documents have tree-like structures, so scrapers can use XPath to navigate through them by selecting nodes according to various parameters. A scraper may combine DOM parsing with XPath to extract whole web pages and publish them on a destination site.

Google Sheets

Google Sheets is a popular tool for data scraping. Scarpers can use the IMPORTXML function in Sheets to scrape from a website, which is useful if they want to extract a specific pattern or data from the website. This command also makes it possible to check if a website can be scraped or is protected.

Language Used for Data Scrapping:

Python is used for the data scrapping , using selenium with python we are accessing the redbus website and accessign the routes by giving the keypoint from selenium, Selenium will automatically access the redbus website and accessing the Xpath values by using webdriver in selenium,

Selenium:

Selenium helps by acting like a real person browsing the website. It can click buttons, enter information, and move through pages like we do. This makes it possible to collect data from websites that change based on user's behavior.

**Step 1: Install Selenium**First, install Selenium using pip:

**“pip install selenium”**

**Step 2: Download WebDriver**  
You'll need a WebDriver for the browser you want to automate (e.g., Chrome, Firefox). For Chrome, download Chrome Driver, Make sure the WebDriver version matches your browser version. Place the WebDriver in a known directory or update the system path.

**from selenium import webdriver**

**driver = webdriver.Chrome()**

**Step 3: Import Selenium and Initialize WebDriver**  
Import Selenium and initialize the WebDriver in your script.

**Step 4: Project running browser**  
Open a website and fetch its content.

use : <https://www.redbus.in/>

**Step 5: Interact with page**

**Step 6: Print content**Now, we can print the content after performing a certain action on the page.

### Data Set Requirements & Explanation:

The scraped dataset for this project should contain detailed information about bus services available on Redbus, covering various aspects critical to travelers and service providers. Here is a breakdown of the fields required:

* **Bus Routes Name:** This field captures the start and end locations of each bus journey, providing crucial information about the routes serviced.
* **Bus Routes Link**: Link for all the route details.
* **Bus Name:** The name of the bus or the service provider, which helps in identifying the specific operator.
* **Bus Type (Sleeper/Seater/AC/Non-AC):** This field specifies whether the bus is a sleeper or seater type, indicating the seating arrangements and comfort level offered.
* **Departing Time:** The scheduled departure time of the bus, essential for planning travel schedules.
* **Duration:** The total duration of the journey from the departure point to the destination, helping passengers estimate travel time.
* **Reaching Time:** The expected arrival time at the destination, allowing for better planning of onward travel or activities.
* **Star Rating:** A rating provided by passengers, indicating the quality of service based on factors such as comfort, punctuality, and staff behavior.
* **Price:** The cost of the ticket for the journey, which can vary based on factors like bus type and demand.
* **Seat Availability:** The number of seats available at the time of data scraping, giving real-time insight into the occupancy levels.

**Source Code and Screen Shots:**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.action\_chains import ActionChains

from time import sleep

service = Service(executable\_path='./chromedriver.exe')

options = webdriver.ChromeOptions()

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://www.redbus.in")

ids = driver.find\_elements("class name",'rtcBack')

coun = 0;

for ii in ids:

coun = coun + 1

print("--")

idsa = ii.find\_elements("tag name", 'li')

print(str(coun),".",ii.find\_element("class name",'rtcName').text)

state\_No = input("Please enter an integer: ")

elem = driver.find\_element("xpath",'//\*[@id="Carousel"]/div['+state\_No+']')

action = ActionChains(driver)

action.move\_to\_element(elem).click().perform()

sleep(5)

ids = driver.find\_element("class name",'DC\_117\_paginationTable')

ids = ids.find\_elements("tag name",'div')

coun = 0;

for ii in ids:

print(ii.text) #//\*[@id="root"]/div/div[4]/div[12]/div[2]

coun = coun + 1

action = ActionChains(driver)

# perform the operation

action.move\_to\_element(ii).click().perform()

sleep(5)

print("----------------------lit of routes")

ids = driver.find\_elements("class name",'route\_link')

coun = 0;

for ii in ids:

linktext = ''+ ii.find\_element("tag name",'a').text

print(ii.find\_element("tag name",'a').get\_attribute('href')," : ",linktext)

pglink = ii.find\_element("tag name",'a').get\_attribute('href')

sleep(10)

driver2 = webdriver.Chrome(service=service, options=options)

driver2.get(pglink)

sleep(10)

mydb = mysql.connector.connect(

host=" 127.0.0.1",

user="root",

password="",

database="redbus"

)

mycursor = mydb.cursor()

last\_height = driver.execute\_script("return document.body.scrollHeight")

scroll\_pause\_time = 1

screen\_height = driver.execute\_script("return window.screen.height;")

i = 1

while True:

driver.execute\_script("window.scrollTo(0, {screen\_height}\*{i});".format(screen\_height=screen\_height, i=i))

i += 1

time.sleep(scroll\_pause\_time)

scroll\_height = driver.execute\_script("return document.body.scrollHeight;")

if (screen\_height) \* i > scroll\_height:

break

sleep(5)

def getdata(dataid):

#div\_element = driver.find\_element("id",'26762666')

#row\_data = driver.find\_element(BY.XPATH,'//\*[@id="'+dataid+'"]/section/div[2]/h1')

#route\_name = row\_data.text

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[1]/div[1]')

bus\_name = row\_data.text

print("BUsname: ", bus\_name)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[1]/div[2]')

bus\_type = row\_data.text

print("Bus Type: ", bus\_type)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[2]/div[1]')

starting\_time = row\_data.text

print("Starting Time: ", starting\_time)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[3]/div')

duration = row\_data.text

print("Duration: ", duration)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[4]/div[1]')

reaching\_time = row\_data.text

print("Reaching Time: ", reaching\_time)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[6]/div') #//\*[@id="24696607"]/div/div[1]/div[1]/div[6]/div

price = row\_data.text

print("Price: ", price)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[7]/div[1]')

seats\_available = row\_data.text

print("Seats Available: ", seats\_available)

sql = "INSERT INTO bus\_routes (busname, bustype, departing\_time, duration, reaching\_time, price, seats\_available) VALUES (%s,%s,%s,%s,%s,%s,%s)"

val = (bus\_name, bus\_type,starting\_time,duration,reaching\_time,price,seats\_available)

print(sql)

print (val)

mycursor.execute(sql, val)

mydb.commit()

mylast\_id = mycursor.lastrowid

print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_----------",str(mylast\_id))

ul = driver.find\_element("class name",'bus-items')

ul = ul.find\_elements("tag name",'li')

coun = 0;

for ii in ul:

if(str(ii.get\_attribute('id'))):

dataid = str(ii.get\_attribute('id'))

getdata(dataid)

**Code For Button Click:**

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.common.keys import Keys

from selenium.common import NoSuchElementException

from selenium.webdriver.common.action\_chains import ActionChains

from selenium.webdriver.support.select import Select

import mysql.connector

from time import sleep

import time

service = Service(executable\_path='./driver/chromedriver.exe')

options = webdriver.ChromeOptions()

driver = webdriver.Chrome(service=service, options=options)

driver.get("https://www.redbus.in/bus-tickets/vijayawada-to-hyderabad?fromCityId=134&toCityId=124&fromCityName=Vijayawada&toCityName=Hyderabad&busType=Any&onward=01-Aug-2024&srcCountry=null&destCountry=null")

sleep(10)

mydb = mysql.connector.connect(

host=" 127.0.0.1",

user="root",

password="",

database="redbus"

)

mycursor = mydb.cursor()

last = driver.find\_element("xpath", '//\*[@id="result-section"]/div[2]/div/div[2]/div/div[4]/div[2]')

last.click()

last\_height = driver.execute\_script("return document.body.scrollHeight")

scroll\_pause\_time = 1 # You can set your own pause time. My laptop is a bit slow so I use 1 sec

screen\_height = driver.execute\_script("return window.screen.height;") # get the screen height of the web

i = 1

while True:

driver.execute\_script("window.scrollTo(0, {screen\_height}\*{i});".format(screen\_height=screen\_height, i=i))

i += 1

time.sleep(scroll\_pause\_time)

scroll\_height = driver.execute\_script("return document.body.scrollHeight;")

if (screen\_height) \* i > scroll\_height:

break

sleep(5)

def getdata(dataid):

#div\_element = driver.find\_element("id",'26762666')

# row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[1]/div[1]')

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[1]/div[1]')

bus\_name = row\_data.text

print("BUsname: ", bus\_name)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[2]/div[1]')

bus\_type = row\_data.text

print("Bus Type: ", bus\_type)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[3]/div')

starting\_time = row\_data.text

print("Starting Time: ", starting\_time)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[4]/div[1]')

duration = row\_data.text

print("Duration: ", duration)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[6]/div/div')

reaching\_time = row\_data.text

print("Reaching Time: ", reaching\_time)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[6]/div/div/span')

#row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[7]/div[1]/span')

price = row\_data.text

print("Price: ", price)

row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[7]/div[1]') #//\*[@id="26704174"]/div/div[1]/div[1]/div[7]/div[1]

#row\_data = driver.find\_element(By.XPATH,'//\*[@id="'+dataid+'"]/div/div[1]/div[1]/div[6]/div/div/span')

seats\_available = row\_data.text

print("Seats Available: ", seats\_available)

sql = "INSERT INTO bus\_routes (busname, bustype, departing\_time, duration, reaching\_time, price, seats\_available) VALUES (%s,%s,%s,%s,%s,%s,%s)"

val = (bus\_name, bus\_type,starting\_time,duration,reaching\_time,price,seats\_available)

print(sql)

print (val)

mycursor.execute(sql, val)

mydb.commit()

mylast\_id = mycursor.lastrowid

ul = driver.find\_element("class name",'bus-items')

ul = ul.find\_elements("tag name",'li')

coun = 0;

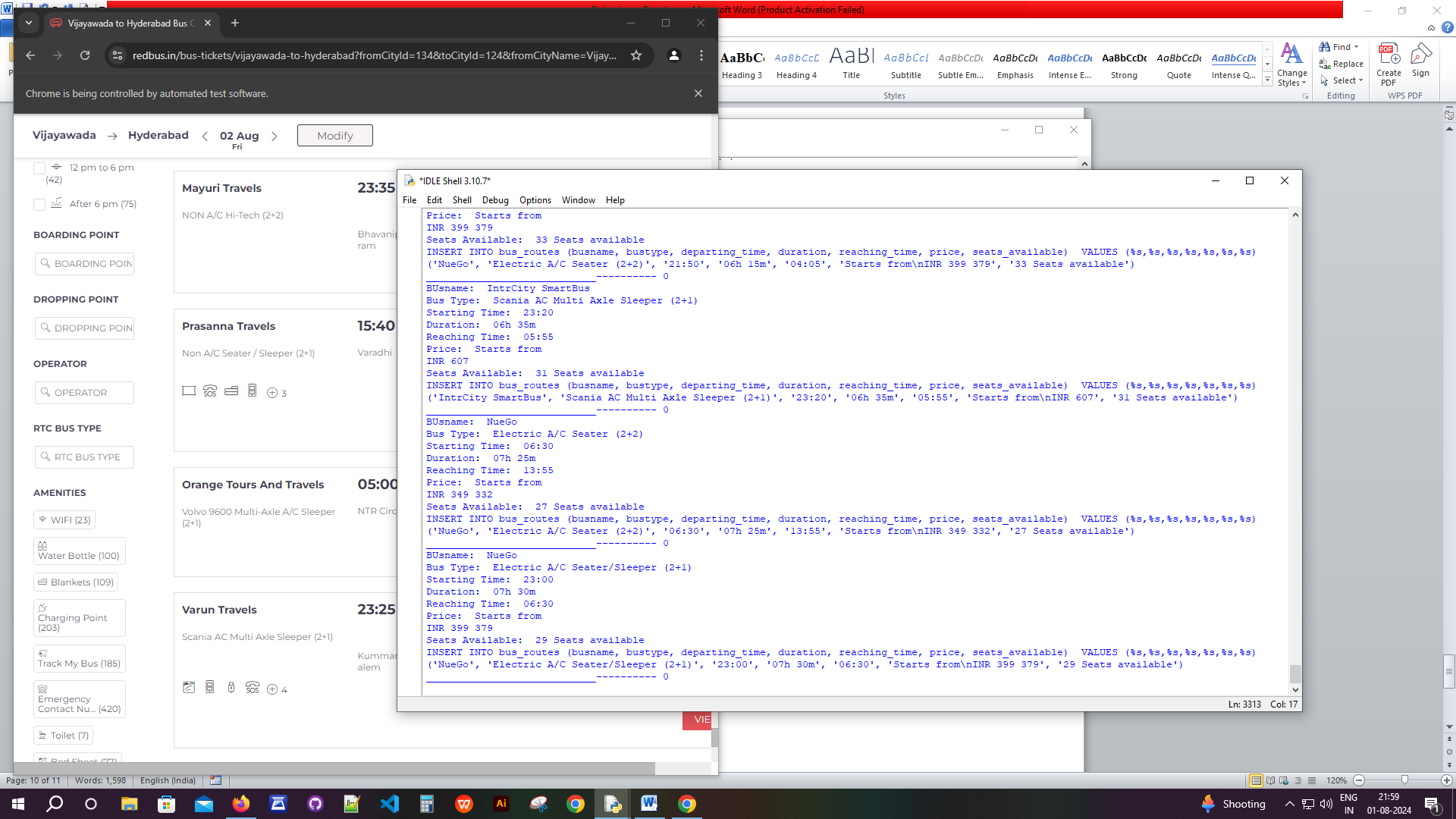
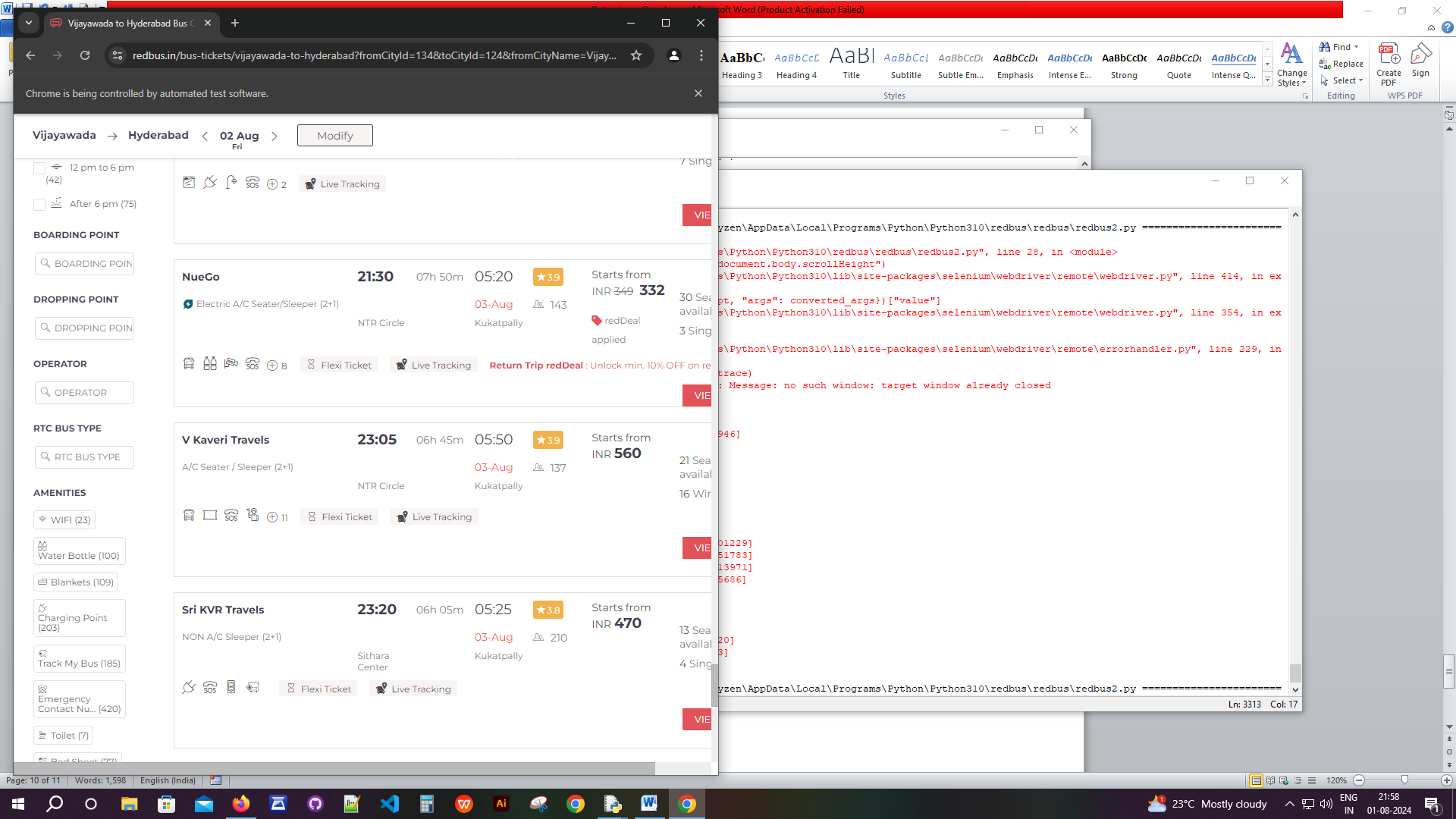
for ii in ul:

if(str(ii.get\_attribute('id'))):

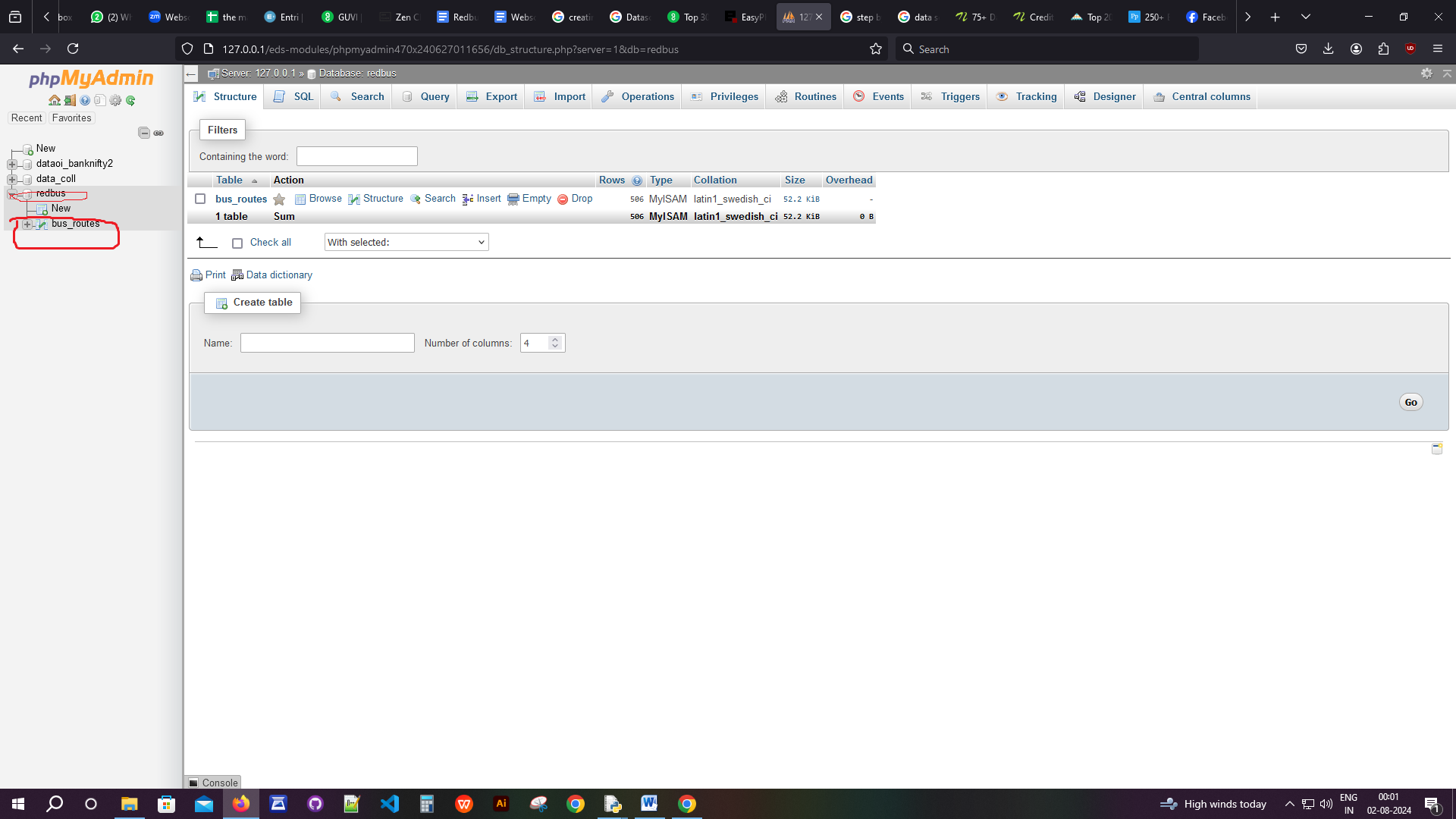
dataid = str(ii.get\_attribute('id'))

getdata(dataid)

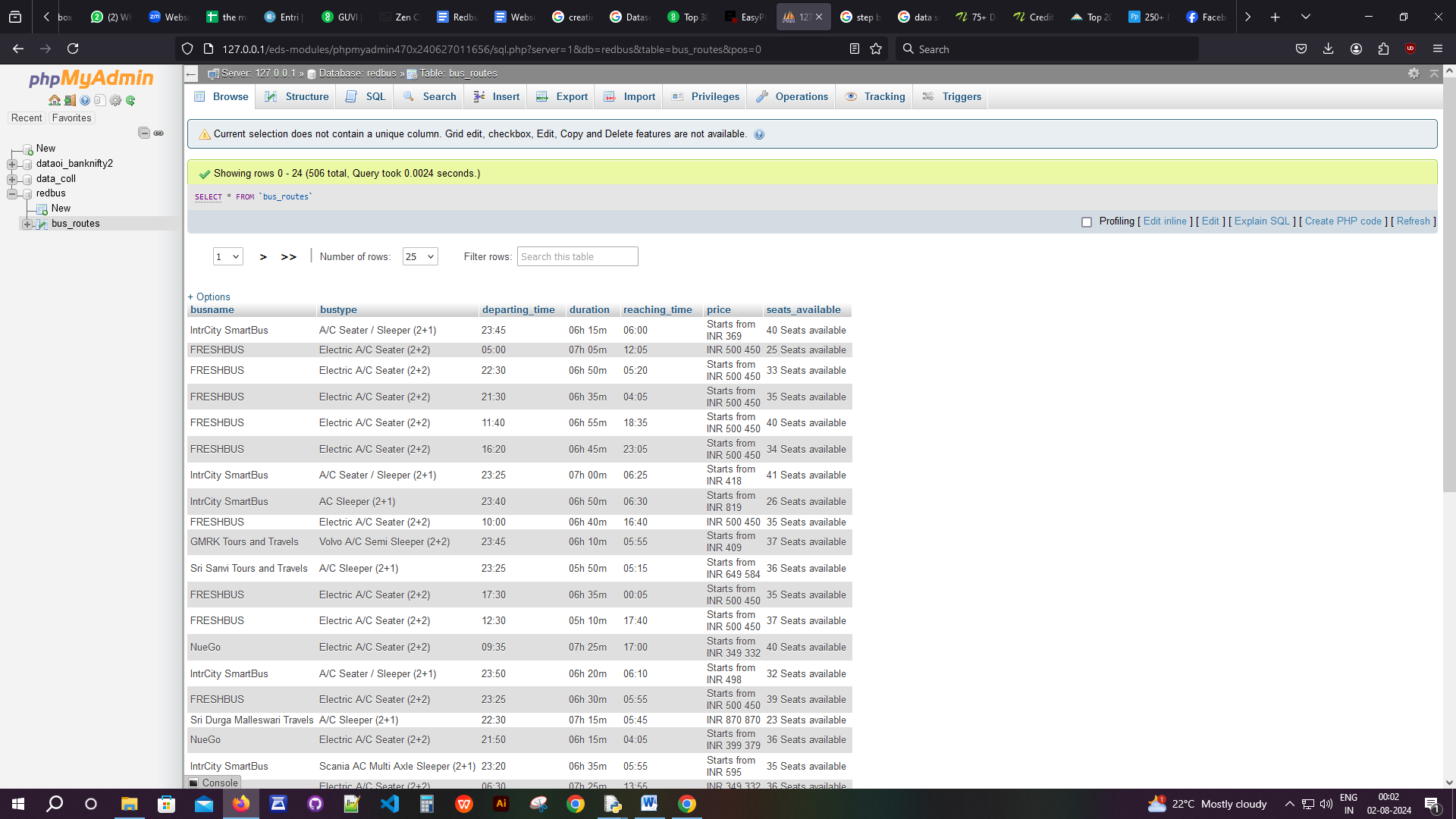
**Screen Shots for Running**



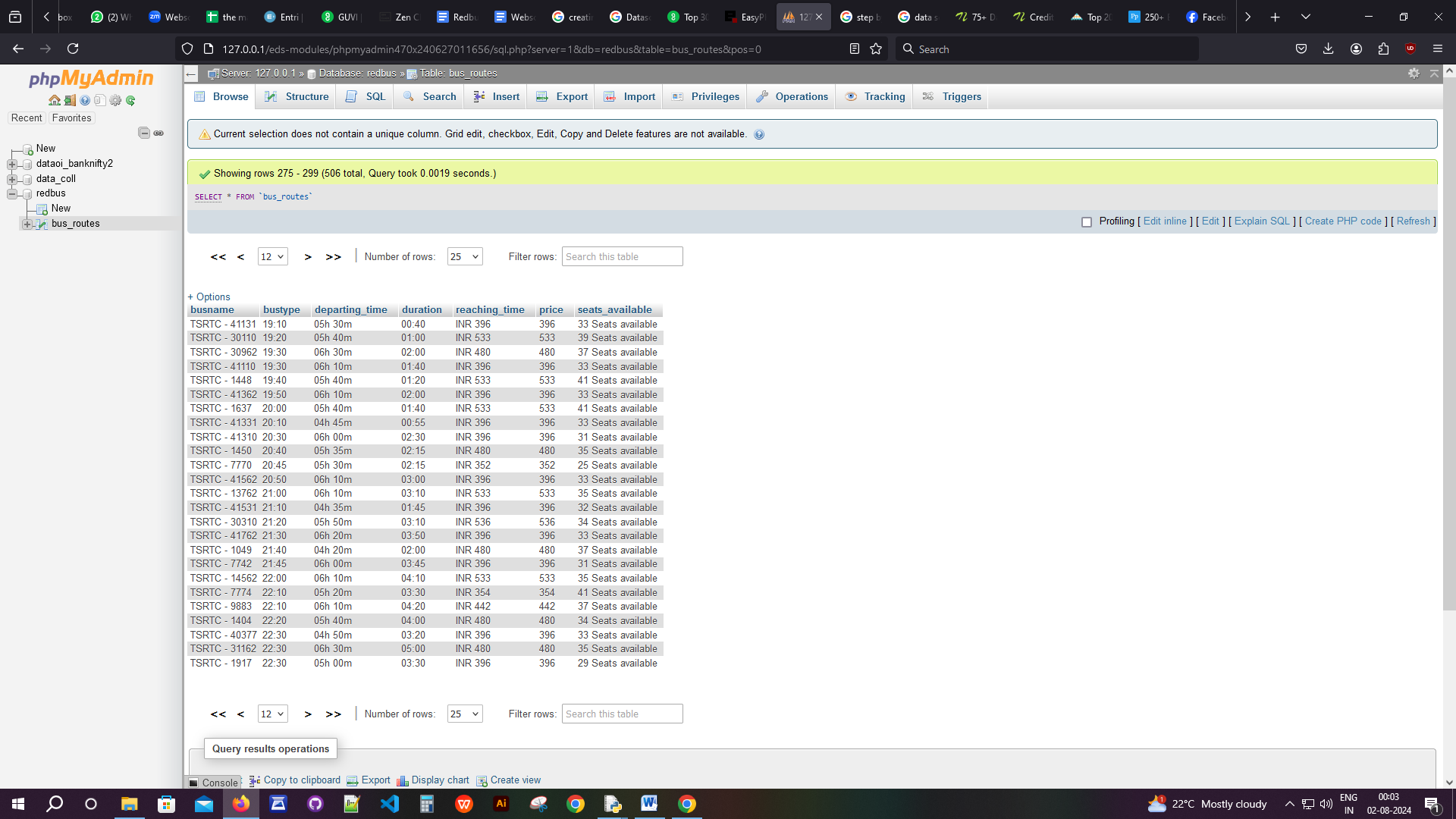
MySQL DB:



Privet Bus Data:



Government Buses:

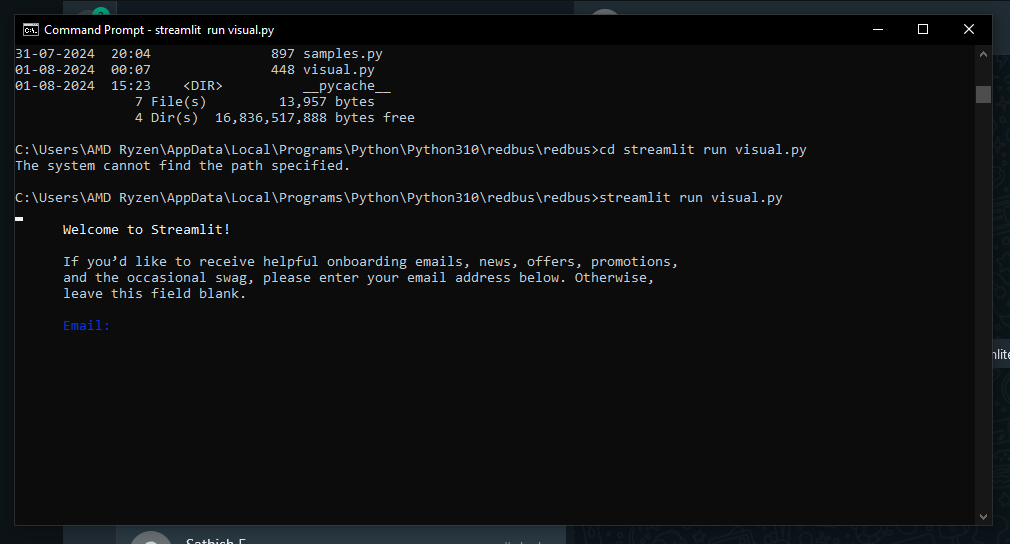


Stream Lit

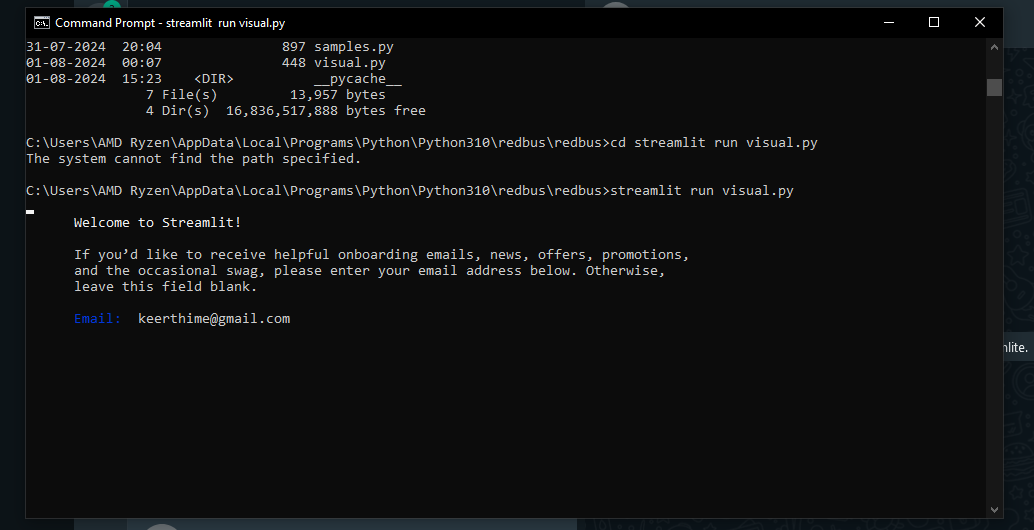
Streamlit is an open-source tool to build and deploy data applications with less coding compared to other front-end technologies like HTML, CSS, and JavaScript. It is a low-code tool specifically designed for building data science applications.

Moreover, the Streamlit library has functions and methods in pure Python to develop data applications with minimal code. Streamlit also supports various database connections, like AWS S3, MS SQL Server, Oracle DB, spreadsheets, etc.

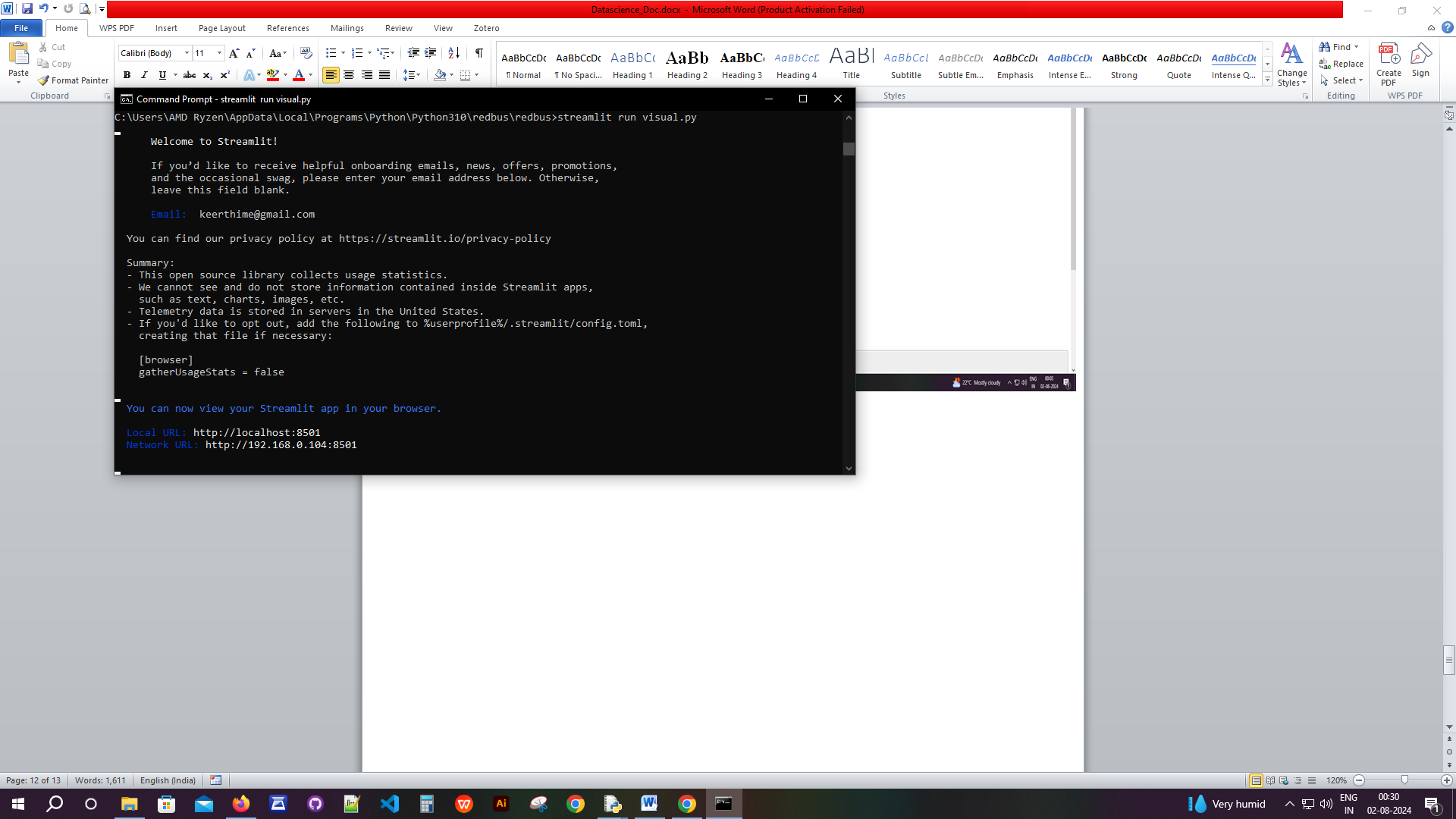
Stream lit configuration and Running Screen



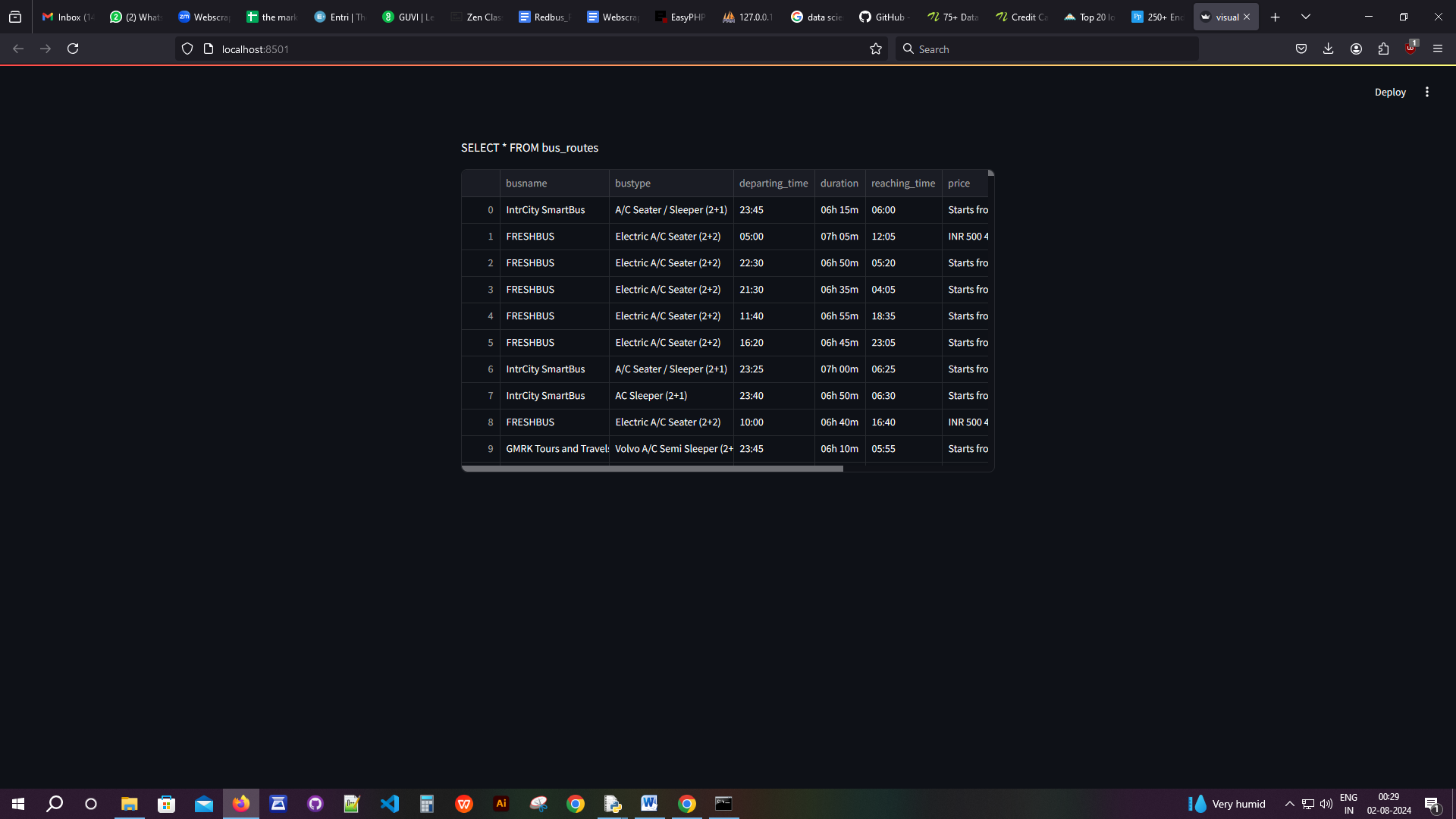
Email Screen for Stream Lit:



After entering the E-Mail ID :



Running Screen for Streamlit:



Conclusion:

Data scraping is a powerful technique that can provide numerous benefits to businesses and individuals alike. Using the right tools and methods, companies can gain insights into their competitors, generate leads, and conduct market research. However, using data scraping responsibly and ethically is essential to avoid legal issues. With the right approach, data scraping can be a valuable asset to any business looking to stay ahead of the curve.